W9_COMPUTER PROGRAMMINING 2019 SPRING

W9. Advanced data structures: Generic methods

We will use the class IO

```
 import java.util.*;
import javax.swing.*;
import java.awt.Font;

class IO
{  static Scanner input = new Scanner( System.in );
   //change font and size for JOptionPane class,//example font "Arial"
   //example size 14
   public static void setOptionPane(String font,int size)
   {UIManager.put("OptionPane.messageFont", new Font(font, Font.PLAIN, size));}
   //array input
   public static double[] Dinput(int n)
   {double c[]=new double[n];
    for(int i=0;i<n;i++)
    {c[i]=Dinput("a["+i+"] = ");}
    return c;
   }

   public static double[][] Dinput(int n,int m)
   {double c[][]=new double[n][m];
    for(int i=0;i<n;i++)
    {for(int j=0;j<m;j++)
     {c[i][j]=Dinput("a["+i+","+j+"] = ");}
    }
    return c;
   }

   public static int[] Iinput(int n)
   {int c[]=new int[n];
    for(int i=0;i<n;i++)
    {c[i]=Iinput("a["+i+"] = ");}
    return c;
   }
   public static int[][] Iinput(int n,int m)
   {int c[][]=new int[n][m];
    for(int i=0;i<n;i++)
    {for(int j=0;j<m;j++)
    {c[i][j]=Iinput("a["+i+","+j+"] = ");}
    }
    return c;
   }
   public static String[] input(int n)
   {String c[]=new String[n];
    for(int i=0;i<n;i++)
    {c[i]=input("a["+i+"] = ");}
    return c;
   }
   public static String[][] input(int n,int m)
   {String c[][]=new String[n][m];
    for(int i=0;i<n;i++)
    {for(int j=0;j<m;j++)
    {c[i][j]=input("a["+i+","+j+"] = ");}
    }
    return c;
   }
   public static String toString(double a[],int n)
   { Locale us=new Locale("us");
     String s1="[";
     for(int i=0;i<a.length;i++)
     { s1+=String.format(us,"%"+n+"f",a[i]);}
     s1+="]\n";
     return s1;
   }

   public static String toString(int a[],int n)
   {String s1="[";
   for(int i=0;i<a.length;i++)
    { s1+=String.format("%"+n+"d",a[i]);}
   s1+="]\n";
   return s1;
   }
```

```
  public static String toString(String a[],int n)
  {String s1="[";
  for(int i=0;i<a.length;i++)
   { s1+=String.format("%"+n+"s",a[i]);}
  s1+="]\n";
  return s1;
  }

  public static String toString(double a[][],int n)
  { String s1="";
    for(int i=0;i<a.length;i++)
    { s1+=toString(a[i],n);}
    return s1;
  }

  public static String toString(int a[][],int n)
  { String s1="";
    for(int i=0;i<a.length;i++)
    { s1+=toString(a[i],n);}
    return s1;
  }

  public static String toString(String a[][],int n)
  { String s1="";
    for(int i=0;i<a.length;i++)
    { s1+=toString(a[i],n);}
    return s1;
  }

  public static void print(String s)
  {JOptionPane.showMessageDialog(null,s);}

  public static void Cprint(String s)
  {System.out.print(s);}

  public static void Cprintln(String s)
  {System.out.println(s);}

 public static double DCinput(String s)
 {     System.out.print(s);
       return Double.parseDouble(input.next());}

 public static int ICinput(String s)
 {   Cprint(s);return input.nextInt();}

 public static String Cinput(String s)
 {     Cprint(s);return input.next();}

 public static double Dinput(String s)
 {   double x=0;
     try{
                x=Double.parseDouble(JOptionPane.showInputDialog(s));
                  } catch(NumberFormatException e) {System.out.println("number format exception");}
     return x;
 }
 public static int Iinput(String s)
 {   int x=0;
     try{
                x=Integer.parseInt(JOptionPane.showInputDialog(s));
                  } catch(NumberFormatException e) {System.out.println("number format exception");}
     return x;
 }

 public static String input(String s)
 {     return JOptionPane.showInputDialog(s);}
}
```

We will use the class IO1

```
class IO1
{  public static String toString(Integer[] x)
   { String s="";
              for(int i=0;i<x.length;i++)
     {s+=" "+x[i];}
      s+="\n";
     return s;
```

```
  }
  public static String toString(Double[] x)
  { String s="";
            for(int i=0;i<x.length;i++)
   {s+=" "+x[i];}
    s+="\n";
   return s;
  }
  public static String toString(String[] x)
  { String s="";
            for(int i=0;i<x.length;i++)
   {s+=" "+x[i];}
    s+="\n";
   return s;
  }
  public static String toString(Character[] x)
  { String s="";
            for(int i=0;i<x.length;i++)
   {s+=" "+x[i];}
    s+="\n";
   return s;
  }
}
```

We will use the class IO2

```
//Generic definition <E>
class IO2
{ public static<E> String toString(E[] x)
  { String s="";
            for(int i=0;i<x.length;i++)
   {s+=" "+x[i];}
    s+="\n";
   return s;
  }
}
```

EX 1: using IO1

```
 public class W9E1
{public static void main(String arg[])
 {Integer[] i={1,2,3,4,5,6};
Double[] d={1.1,2.2,3.3,4.4,5.5,6.6};
Character[] c={'a','b','c','d','f','g'};
String[] s={"ali","veli","49","elli"};
String s1="";
s1+=IO1.toString(i);
s1+=IO1.toString(d);
s1+=IO1.toString(c);
s1+=IO1.toString(s);
IO.Cprint(s1);
IO.print(s1);
}
}
```

EX2: using IO2

```
public class W9E2
{public static void main(String arg[])
 {Integer[] i={1,2,3,4,5,6};
Double[] d={1.1,2.2,3.3,4.4,5.5,6.6};
Character[] c={'a','b','c','d','f','g'};
String[] s={"ali","veli","49","elli"};
String s1="";
s1+=IO2.toString(i);
s1+=IO2.toString(d);
s1+=IO2.toString(c);
s1+=IO2.toString(s);
IO.Cprint(s1);
IO.print(s1);
}
}
```

EX3: Classes with generic variables

```
public class matrix<E>
```

```
{
public E[][] A;
//constructor methods
public matrix(E[][] Ai)
{   try{
              @SuppressWarnings("unchecked")
              final E[][] A1 =(E[][]) new Object[Ai.length][Ai[0].length];
              A=A1;
   }catch(EmptyMatrixException e){}
    catch(FullMatrixException f){};
 input_matrix(Ai);
}

public matrix(matrix<E> M)
{
              try{
             @SuppressWarnings("unchecked")
             final E[][] A1=(E[][])new Object[M.A.length][M.A[0].length];
             A=A1;
     }catch(EmptyMatrixException e){}
    catch(FullMatrixException f){};
              input_matrix(M.A);
}

public void input_matrix(E[][] Ai)
{   for(int i=0;i<A.length;i++)
              { for(int j=0;j<A[0].length;j++)
                {A[i][j]=Ai[i][j];}
   }
}

public String toString()
{
 int n=A.length;
 int m=A[0].length;
 String s1="";
 for(int i=0;i<n;i++)
 {  for(int j=0;j<m;j++)
  { s1+=String.format(" %s ",A[i][j]);}
 s1+="\n";
 }
return s1;
} //end of method toString

}//end of class matrix
```

```
public class EmptyMatrixException extends RuntimeException
{private static final long serialVersionUID = 2097498L;
              public EmptyMatrixException()
 {super("Matrix is empty");}
public EmptyMatrixException(String exception)
 {super(exception);}
 }
```

```
public class FullMatrixException extends RuntimeException
{private static final long serialVersionUID = 547638L;
              public FullMatrixException()
 {super("Matrix is full");}
public FullMatrixException(String exception)
 {super(exception);}
 }
```

```
public class W9E3
{public static void main(String arg[])
{Double[][] a={{1.1,2.2},{3.3,4.4}};
 Integer[][] b={{1,2},{3,4}};
 Character[][] c={{'a','b'},{'c','d'}};
 matrix<Double> A=new matrix<Double>(a);
 matrix<Integer> B=new matrix<Integer>(b);
 matrix<Character> C=new matrix<Character>(c);
 String s=""+A+B+C;
 IO.print(s);
 }
}
```

EX4

USING LAMBDA EXPRESSIONS WITH GENERIC CLASSES

Using of Lambda expressions are introduced. In java several built in Functional interfaces are defined to be specifically used as Lambda expressions. Some of the most common interfaces are listed in the table below.

Table 14.3 Important functional interfaces in Java

Interface name	Argument	Returns
Predicate<T>	T	boolean
Consumer<T>	T	void
Function<T,R>	T	R
BiFunction<T,W,R>	(T,W)	R
Supplier<T>	None	T
UnaryOperator<T>	T	T
BinaryOperator<T>	(T,T)	T
Binary Function<T,W,R>	(T,W)	R

```
import java.util.function.*;
public class  W9E4a
{ public static void main(String[] arg)
  { Function<Double,Double> y = x -> {return x*x-2.0*+5;};
    IO.print(""+y.apply(2.0));
  }
}
```

```
import java.util.function.*;
public class  W9E4b
{ public static void main(String[] arg)
  { Function<Double,Double> y = x -> {return x*Math.sin(x);};
    IO.print(""+y.apply(2.0));
  }
}
```

```
 import java.util.function.*;
public class  W9E4c
{ public static void main(String[] arg)
  { Function<Double,Boolean> y = x -> (x*x-2.0*x+5)>2;
    IO.print(""+y.apply(2.0));
  }
}
```

```
import java.util.function.*;
public class  W9E4d
{ public static void main(String[] arg)
  { BiFunction<Double,Double,Boolean> z = (x,y) -> (x*x+y*y)>3;
    IO.print(""+z.apply(2.0,3.0));
  }
}
```

```
import java.util.function.*;
public class  W9E4e
{ public static void main(String[] arg)
  { BiFunction<Double,Double,Double> z = (x,y) -> (x*x+y*y);
    IO.print(""+z.apply(2.1,Math.PI));
  }
}
```

EX5

```
 import java.util.function.*;
 public class W9E5
```

```
{
  public static double bisection(Function<Double,Double> f,double a,double b)
  {double b1=1.1*b;
   double r=(a+b)/2.0;
   double eps=1.0e-8;
   int nmax=100;
   int i=0;
   while(Math.abs(f.apply(r))>eps && i<nmax)
   {if(f.apply(a)*f.apply(r)<0) b=r;
    else a=r;
    r=(a+b)/2.0;;
    i++;
   }
   if(i>=nmax) r=bisection(f,a,b1);
   return r;
  }
  public static void main(String arg[])
  { //root of a function
    double a=IO.Dinput("a=");
    double b=IO.Dinput("b=");
    Function<Double,Double> f=x->x*x-2.3*x-2.0;
    double x0=bisection(f,a,b);
    String s="x0="+x0;
    IO.print(s);
  }
}
```

HOMEWORK EXERCISES

Homework exercises will be done at home and will bring to next Thursday class printed no late exercises will be excepted. Each code should include student name id#, code plus results should be given. Homeworks will be accepted in written format plus a computer copy in pdf format will be sent to computer_programming@turhancoban.com adress your file name should be “group”+“week#”+studentname+studentid#.pdf

A W1_turhan_coban_0101333.pdf

B W3_ali_veli_02335646.pdf

W9HW1 : By changing W9E5 find root of function $f(x) = e^x - \pi$

W8HW2 Remember W7E5, It was given as:
EX5 Lambda variables

```
interface if_xy
{public double func(double x[]);}
```

```
public class W7E5
{
  public static void main(String arg[])
  { //root of a function
    double x=IO.Dinput("x=");
    double y=IO.Dinput("y=");
    if_xy f=(double z[])->z[0]*z[0]+z[1]*z[1];
    double z[]={x,y};
    String s="x="+z[0]+"y="+z[1]+"\nx*x+y*y = "+f.func(z);
    IO.print(s);
  }
}
```

Now change this program so that instead of using interface **if_xy**, use interface **BiFunction<Double,Double,Double>** to get the same result.

W8HW3 Use **BiFunction<Double,Double,Booelan>** interface to check if **(x*x+y*x)<5**